AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) An automated computerized method for optimizing allocation of a set *W* of *n* tasks to *m* ~~set W of n tasks to m~~ available resources for accomplishing such tasks using combinatorial multimodal optimization for finding multiple optimal ways of dividing said set *W* of *n tasks* ~~task values~~ into *m* respective ~~respectively~~ groups associated with said available resources, such that each of the groups satisfies a respective constraint condition, said method comprising: ~~the method including execution of~~

using at least one computer to execute a computer program to automatically perform a series of machine operations ~~comprising~~:

(a) receiving digital data signals representing plural tasks for assignment to available resources and, based thereon, defining an initial population of trial solutions assigning specific tasks to specific resources;

(b) calculating for each trial solution a fitness vector comprising *m* elements, each of which is indicative of whether the constraint condition of a corresponding respective one of the *m* groups has been satisfied by the trial solution;

(c) selecting a plurality of trial solutions for a next generation in dependence upon their respective fitness vectors;

(d) creating a new population of trial solutions including the selected earlier trial solutions;

(e) repeating steps (b) to (d) until the population of trial solutions stabilizes, the individual trial solutions of the stable population representing multiple optional ways of dividing the set *W* of *n* tasks; ~~tasks;~~ and

(f) outputting data representing at least one of the individual trial solutions of said stabilized population as an optimized allocation of tasks to resources.

Claim 2 (Canceled).

3. (Previously Presented) A method as in claim 1 in which the fitness vector comprises *m* bits, each bit being indicative of whether the constraint condition of a corresponding one of the *m* groups has been satisfied.

4. (Previously Presented) A method as in claim 1 including calculating a fitness value for each individual trial solution.

5. (Previously Presented) A method as claimed in claim 3 including calculating a fitness value for each individual trial solution in which the fitness value comprises the sum of the bits in the fitness vector.

6. (Previously Presented) A method as in claim 1 including reserving a proportion of the new population for individual trial solutions selected at step (c).

7. (Previously Presented) A method as 6 in which a non-reserved proportion of the new population is generated using a Roulette wheel selection method.

8. (Previously Presented) A method as in claim 1 in which step (c) comprises selecting non-dominated individual trial solutions using the criteria of Pareto optimality.

9. (Previously Presented) A method as in claim 4 in which step (c) comprises selecting non-dominated individual trial solutions using the criteria of Pareto optimality including ranking non-dominated individual trial solutions by fitness value, and selecting from the ranked list.

10. (Previously Presented) A method as in claim 9 in which only non-dominated individual trial solutions with greatest fitness value may be selected at step (c).

11. (Previously Presented) A method as in claim 4 in which step (c) comprises selecting individual trial solutions in dependence upon both their respective fitness vectors and their respective fitness values.

12. (Previously Presented) A method as in claim 1 in which crossover and mutation are applied at step (d) to at least some individual trial solutions in the new population.

13. (Previously Presented) A method as in claim 1 in which step (c) comprises selecting no more than one individual trial solution for each unique fitness vector.

14. (Currently Amended) An automated computerized method of distributing a plurality of tasks between a plurality of devices connected together to form a network, wherein each device has an associated constraint on the amount of tasks that it can perform per unit of time, said method comprising: ~~the method including execution of~~ using at least one computer to execute a computer program to automatically perform a series of machine operations ~~comprising~~:

(a) generating a plurality of trial solution allocations of tasks to devices to form an initial population of allocations;

(b) calculating for each trial solution a fitness vector comprising a plurality of elements each of which is ~~indicative of whether the~~ associated ~~constraint of~~ with a corresponding respective one of the plurality of devices and is indicative of whether the constraint associated with that corresponding respective device has been satisfied by the trial solution independently of the extent to which the constraints associated with the other devices have been satisfied by the trial solution;

(c) selecting a plurality of allocations of tasks to devices for inclusion in a next generation of allocations in dependence upon their respective fitness vectors;

(d) creating the next generation of allocations of tasks to devices by including the allocations selected in step (c) together with new allocations, each of which is formed from a combination of two or more of the allocations selected in step (c);

(e) repeating steps (b) to (d) until the population stabilizes; and

(f) outputting data representing an allocation of the tasks among the devices according to one of the allocations included in the stabilized population.

15. (Previously Presented) A method as in claim 14 wherein the devices are processors within a multi-processor computer system.

16. (Previously Presented) A method as in claim 14 wherein the devices are computers within a computer network.

17. (Previously Presented) A method as in claim 14 wherein the devices are routers and the tasks are estimated volumes of traffic to be routed through the routers within a data network, and wherein the allocations are used to form a routing strategy.

18. (Previously Presented) A method as in claim 14 in which step (c) comprises selecting non-dominated allocations using the criteria of Pareto optimality of the associated fitness vectors.

19. (Previously Presented) A method as in claim 1 in which new allocations are formed in step (d) by performing crossover operations in respect of groups of two or more of the allocations selected in step (c).

20. (Previously Presented) A method as in claim 14 in which mutation operations are applied to one or more of the new allocations formed in step (d) according to a predetermined probability of each new allocation being mutated.

21. (Currently Amended) A tangible medium containing a computer program which, when executed effects a method for optimizing allocation of a set *W* of *n* tasks to *m* ~~set W of n tasks to m~~ available resources for accomplishing such tasks using combinatorial multimodal optimization for finding multiple optimal ways of dividing said

set *W* of *n* tasks into *m* ~~set W of n task values into m~~ resource groups, such that each of the groups satisfies a respective constraint condition, said method comprising: ~~the method including execution of~~

using at least one computer to execute a computer program to automatically perform a series of machine operations ~~comprising~~:

(a) defining an initial population of trial solutions;

(b) calculating for each trial solution a fitness vector comprising *m* elements, each of which is indicative of whether the constraint condition of a corresponding respective one of the *m* groups has been satisfied by the trial solution;

(c) selecting a plurality of trial solutions for a next generation in dependence upon their respective fitness vectors;

(d) creating a new population of trial solutions including the selected earlier trial solutions;

(e) repeating steps (b) to (d) until the population of trial solutions stabilizes, the individual trial solution of the stable population representing multiple optional ways of dividing the set *W* of *n* ~~set W of~~ tasks; and

(f) outputting data representing at least one of said stabilized population as an optimized allocation of tasks to resources.

Claim 22 (Canceled).

23. (Previously Presented) A system comprising a plurality of devices connected together to form a network, wherein each device has an associated

constraint on the amount of tasks that it can perform per unit of time, the system including an allocation subsystem for allocating a plurality of tasks among the devices, the allocation subsystem comprising:

(a) means for generating a plurality of trial solution allocations to form an initial population of allocations;

(b) means for calculating for each trial solution allocation a fitness vector comprising a plurality of elements each of which is ~~indicative of whether the~~ associated ~~constraint of~~ with a corresponding respective one of the plurality of devices and is indicative of whether the constraint associated with that corresponding respective device has been satisfied by the trial solution independently of the extent to which the constraints associated with the other devices have been satisfied by the trial solution;

(c) means for selecting a plurality of allocations for inclusion in a next generation of allocations in dependence upon their respective fitness vectors;

(d) means for creating the next generation of allocations by including the allocations selected in step (c) together with new allocations each of which is formed from a combination of two or more of the allocations selected in step (c);

(e) means for repeating steps (b) to (d) until the population stabilizes; and

(f) means for outputting an allocation of the tasks among the devices according to one of the allocations included in the stabilized population.

24. (Currently Amended) A method of operating a multi-processor computer system to execute a computer program including a set of multiple separate tasks which

must all be completed in order for the program execution to be complete, said method comprising:

distributing multiple of said set of program tasks between multiple computer program processor devices to efficiently accomplish all such distributed tasks wherein each computer program processor device has an associated constraint on the amount of tasks that it can perform per unit of time, said distribution of tasks to said processor devices being accomplished by:

(a) receiving digital data signals representing a set of plural tasks for assignment to available processor devices and, based thereon, defining an initial population of trial solutions assigning specific tasks to specific processor devices;

(b) calculating for each trial solution a fitness vector comprising a plurality of elements each of which is ~~indicative of whether the constraint of~~ associated with a corresponding respective one of the multiple computer program processor devices and is indicative of whether the constraint associated with that corresponding respective computer program processor device has been satisfied by the trial solution independently of the extent to which the constraint associated with the other computer program processor devices have been satisfied by the trial solution;

(c) selecting a plurality of trial solutions for a next generation in dependence upon their respective fitness vectors;

(d) creating a new population of trial solutions including the selected earlier trial solutions;

(e) repeating steps (b) to (d) until the population of trial solutions stabilizes, the individual trial solutions of the stable population representing multiple optional ways of dividing the input set of tasks; and

(f) outputting task assignments to said processor devices in conformance with at least one of said stabilized population as an optimized allocation of tasks to resources.

25. (Currently Amended) A multi-processor computer system for executing a computer program including a set of multiple separate tasks which must all be completed in order for the program execution to be complete, said system comprising:

a plurality of computer program processors; and

means networked with said multiple computer program processors for distributing multiple of said set of program tasks between said multiple computer program processor devices to efficiently accomplish all such distributed tasks wherein each computer program processor device has an associated constraint on the amount of tasks that it can perform per unit of time, said distribution of tasks to said processor devices being accomplished by:

(a) receiving digital data signals representing a set of plural tasks for assignment to available processors and, based thereon, defining an initial population trial solutions assigning specific tasks to specific processors;

(b) calculating for each trial solution a fitness vector comprising a plurality of elements each of which is ~~indicative of whether the constraint of~~ associated with a corresponding respective one of the multiple computer program processor devices and

is indicative of whether the constraint associated with that corresponding computer program processor device has been satisfied by the trial solution independently of the extent to which the constraints associated with the other computer program processor devices have been satisfied by the trial solution;

(c) selecting a plurality of trial solutions for a next generation in dependence upon their respective fitness vectors;

(d) creating a new population of trial solutions including the selected earlier trial solutions;

(e) repeating steps (b) to (d) until the population of trial solutions stabilizes, the individual trial solutions of the stable population representing multiple optional ways of dividing the input set of tasks, and

(f) outputting task assignments to said processors in conformance with at least one of said stabilized population as an optimized allocation of tasks to resources.